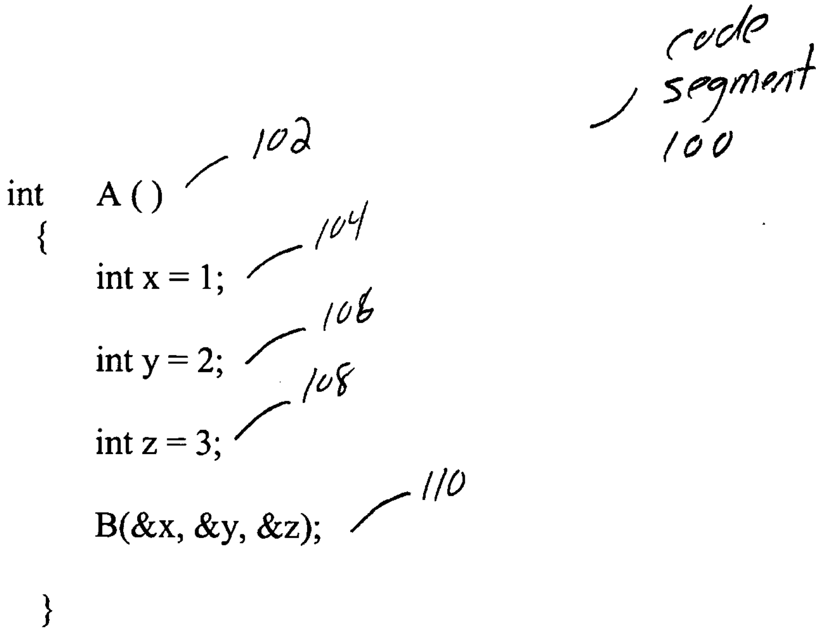


```
int    A ()
  {
       int x = 1;

       int y = 2;

       int z = 3;

       B(&x, &y, &z);

  }
```

FIG. 1

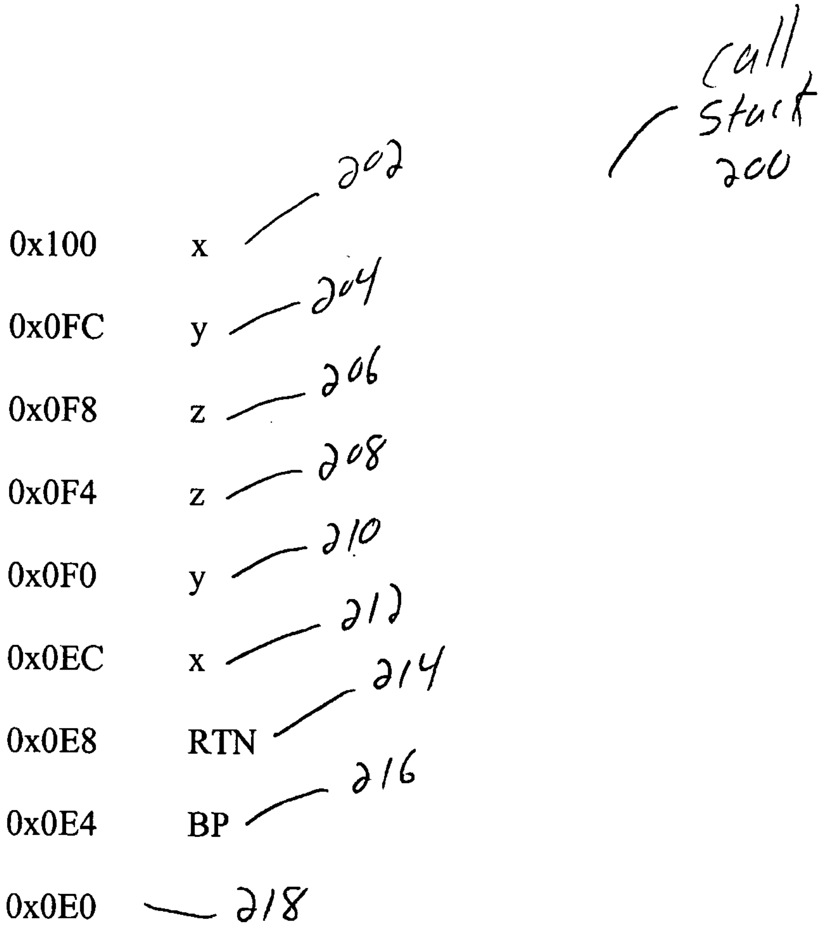


FIG. 2 (Prior Art)

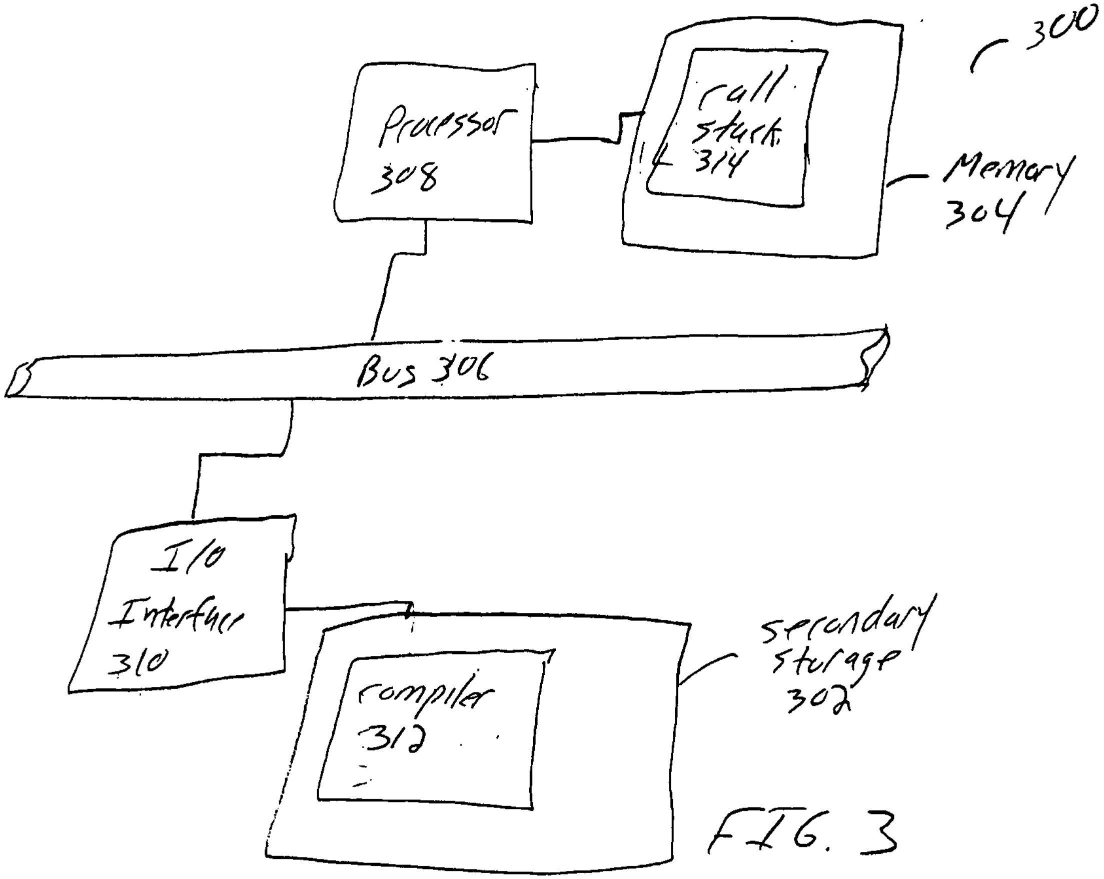
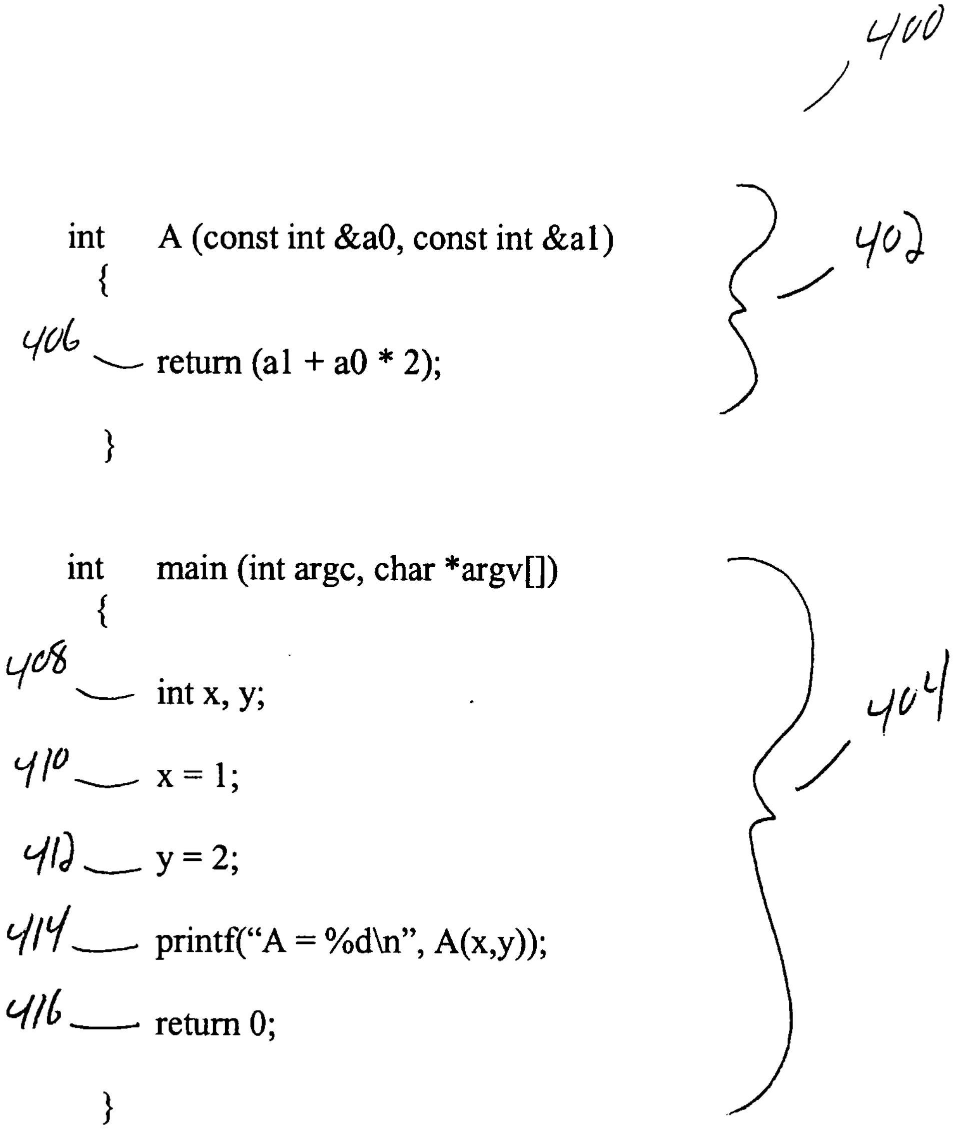

300
Processor 308
call stack 314
Memory 304
Bus 306
I/O Interface 310
compiler 312
secondary storage 302

FIG. 3



```
int     A (const int &a0, const int &a1)
  {

        return (a1 + a0 * 2);

  }

int     main (int argc, char *argv[])
  {

        int x, y;

        x = 1;

        y = 2;

        printf("A = %d\n", A(x,y));

        return 0;

  }
```

FIG. 4

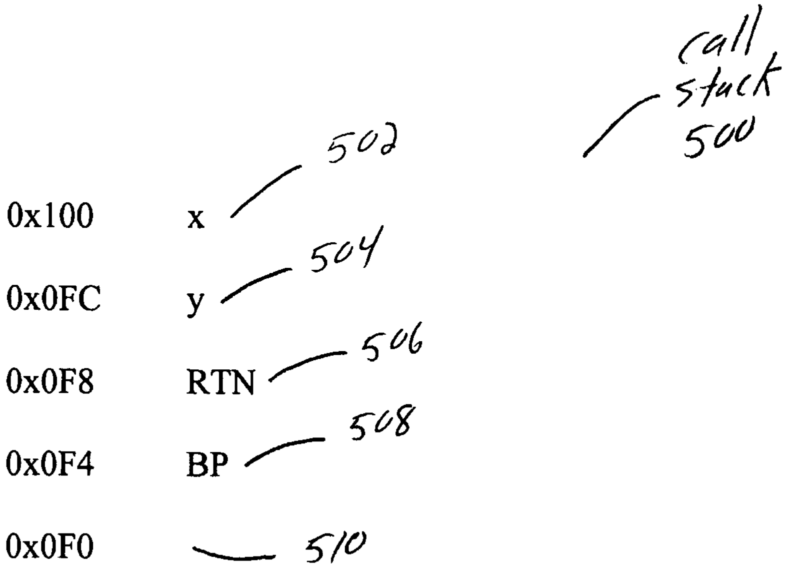
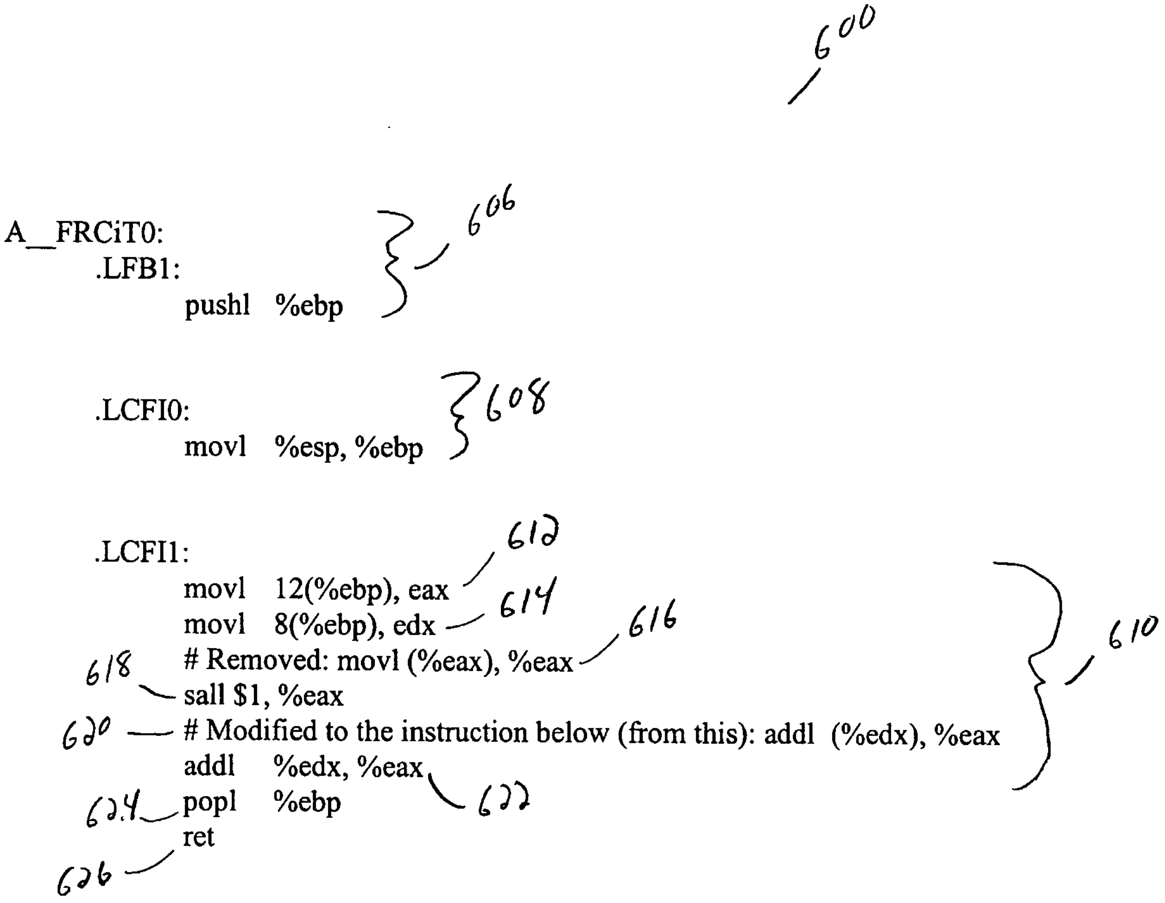

call stack 500
502
0x100 x
504
0x0FC y
506
0x0F8 RTN
508
0x0F4 BP
0x0F0
510

FIG. 5

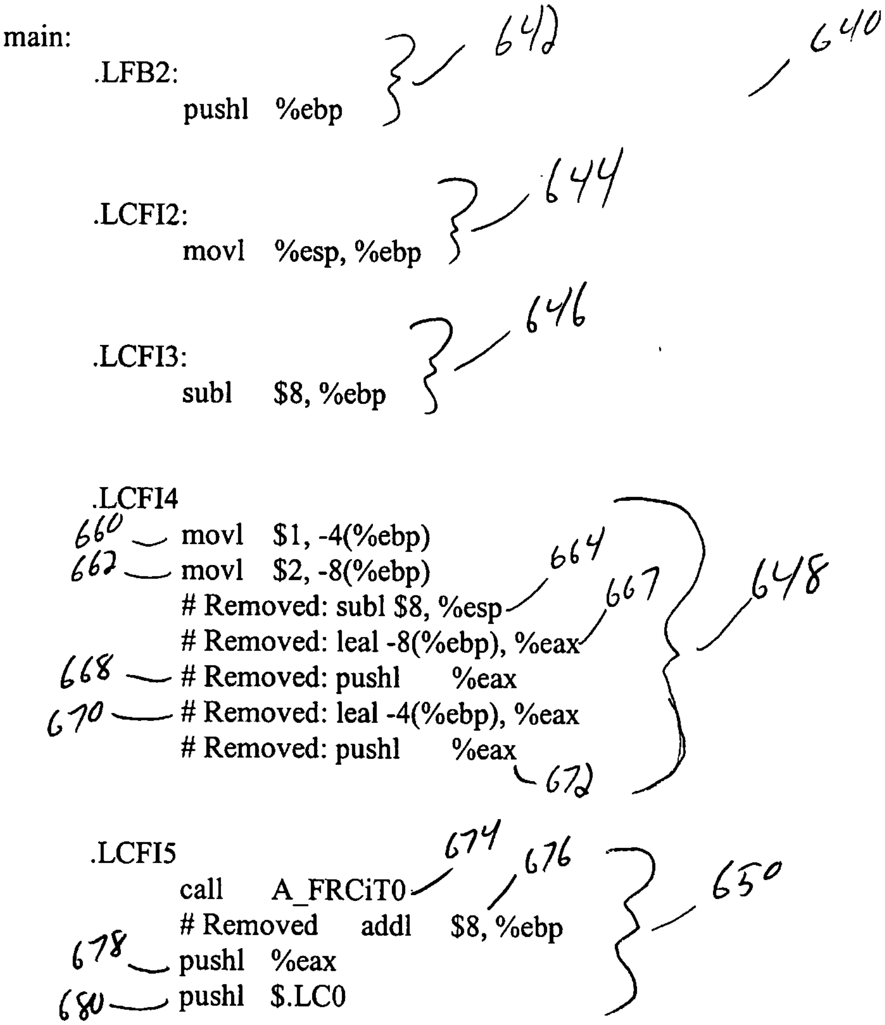


```
A__FRCiT0:
        .LFB1:
                pushl   %ebp


        .LCFI0:
                movl    %esp, %ebp


        .LCFI1:
                movl    12(%ebp), eax
                movl    8(%ebp), edx
                # Removed: movl (%eax), %eax
                sall $1, %eax
                # Modified to the instruction below (from this): addl  (%edx), %eax
                addl    %edx, %eax
                popl    %ebp
                ret
```

FIG. 6A



```
main:
        .LFB2:
                pushl   %ebp

        .LCFI2:
                movl    %esp, %ebp

        .LCFI3:
                subl    $8, %ebp

        .LCFI4
                movl    $1, -4(%ebp)
                movl    $2, -8(%ebp)
                # Removed: subl $8, %esp
                # Removed: leal -8(%ebp), %eax
                # Removed: pushl      %eax
                # Removed: leal -4(%ebp), %eax
                # Removed: pushl      %eax

        .LCFI5
                call    A_FRCiT0
                # Removed     addl     $8, %ebp
                pushl   %eax
                pushl   $.LC0
```

FIG. 6B

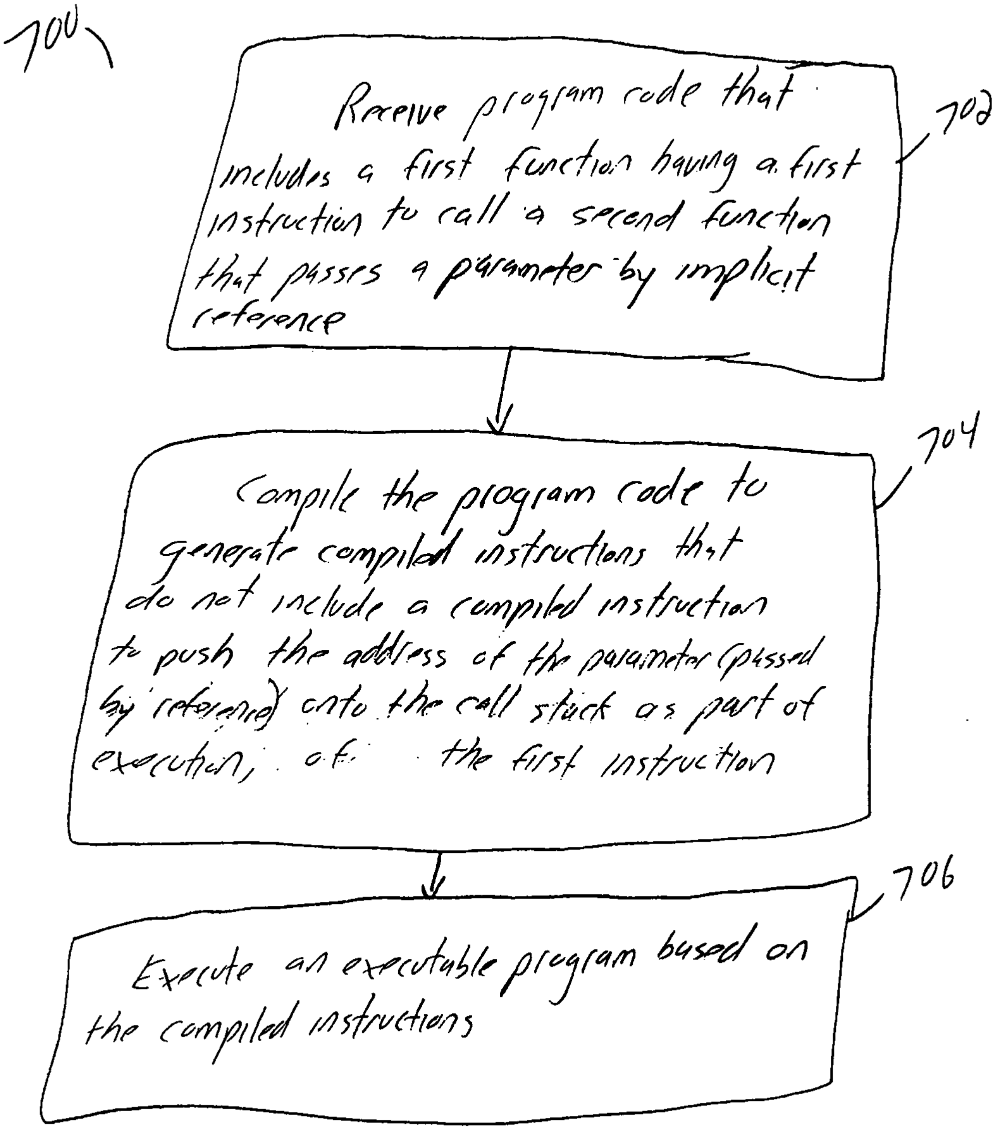

700
Receive program code that includes a first function having a first instruction to call a second function that passes a parameter by implicit reference
702
Compile the program code to generate compiled instructions that do not include a compiled instruction to push the address of the parameter (passed by reference) onto the call stack as part of execution of the first instruction
704
Execute an executable program based on the compiled instructions
706

FIG. 7

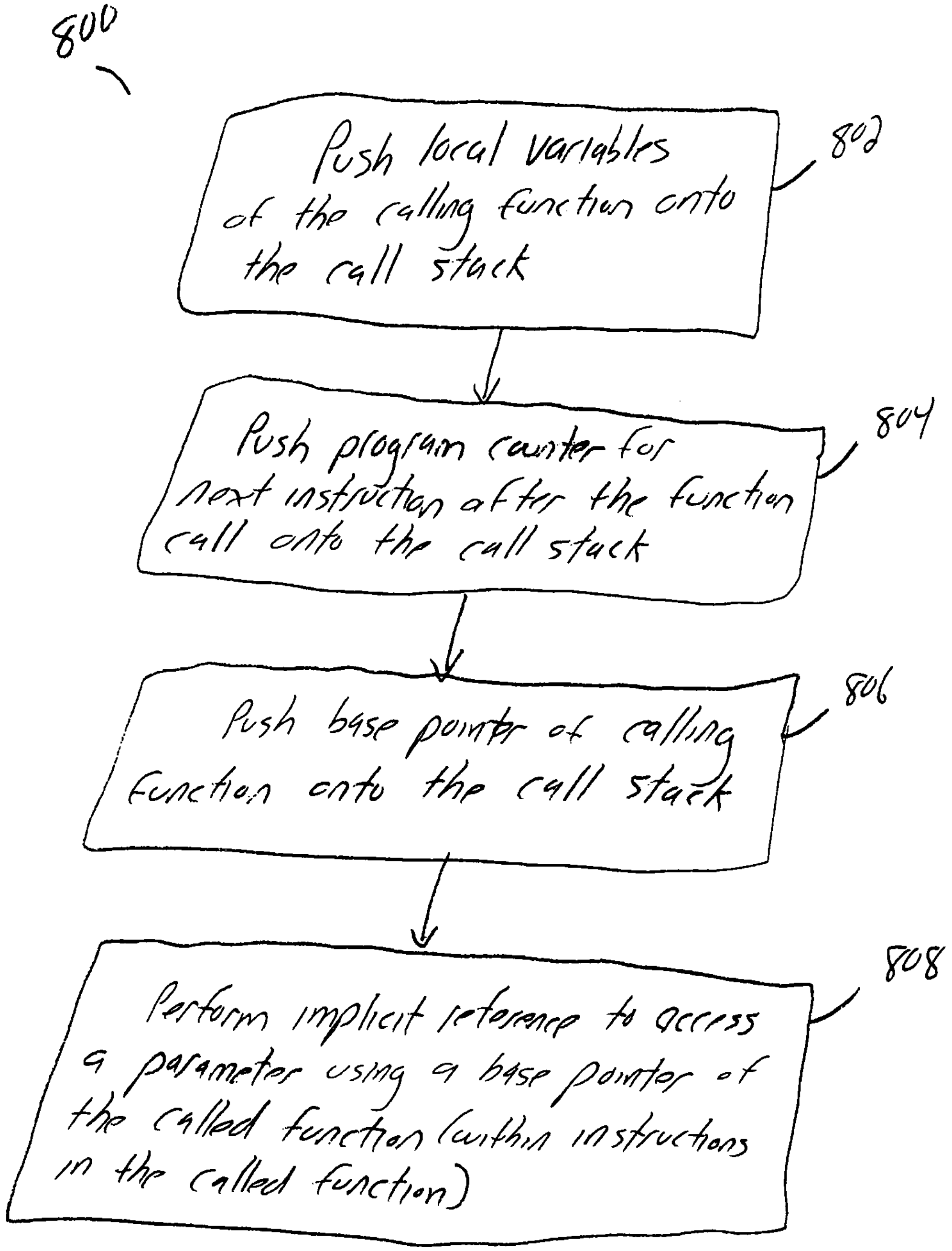

800
Push local variables of the calling function onto the call stack
802
Push program counter for next instruction after the function call onto the call stack
804
Push base pointer of calling function onto the call stack
806
Perform implicit reference to access a parameter using a base pointer of the called function (within instructions in the called function)
808

FIG. 8

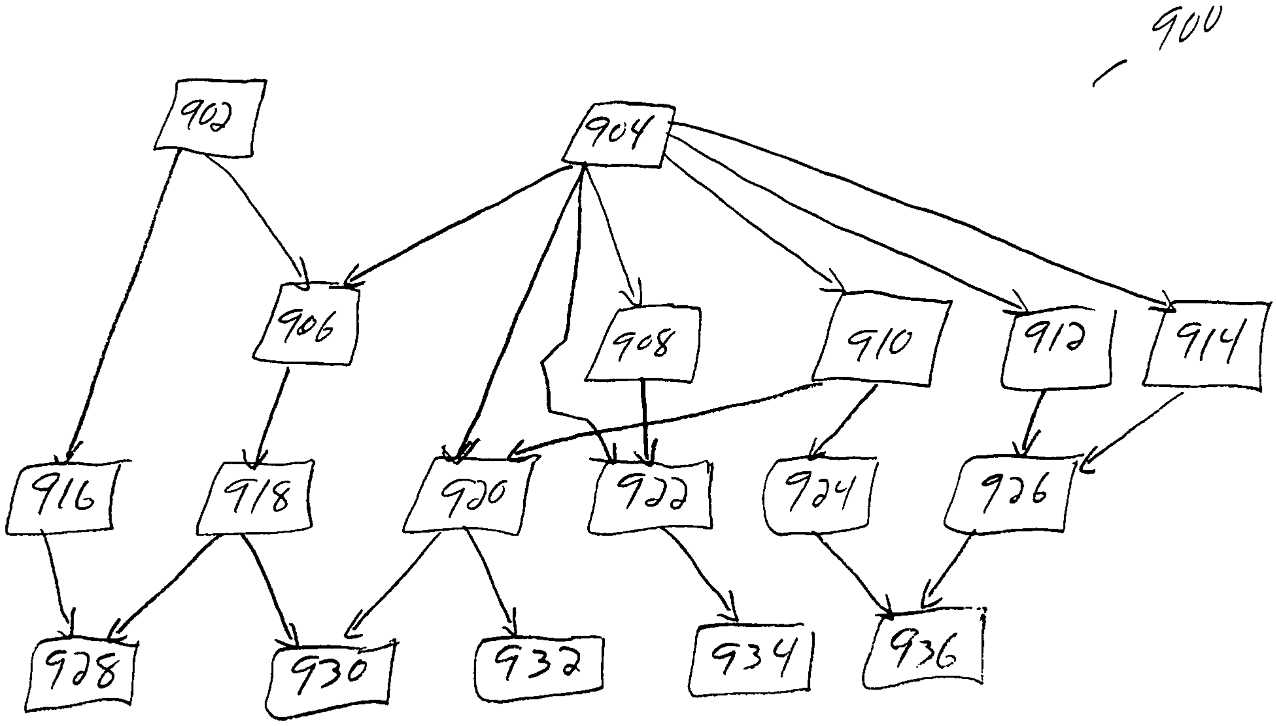


FIG 9